International Journal of Innovative Research in Advanced Engineering (IJIRAE) ISSN: 2349-2163
Volume 1 Issue 5 (June 2014) http://ijirae.com

An Efficient Approach for Asymmetric Data Classification

R.Sateesh Kumar*
Assistant Professor,
vasavi College of Engineering

Y.Anitha
Assistant Professor
vasavi College of Engineering

Abstract - In many classification problems, the number of targets (eg. intruders) present is very small compared with the number of clutter objects. Traditional classification approaches usually ignore this class imbalance, causing performance to experience low accordingly. In contrary, the algorithm considerably imbalanced logistic regression (IILR) algorithm explicitly addresses class imbalance in its formulation. I am proposing this algorithm and give the details necessary to employ it for intrusion detection data sets characterized by class imbalance.

Keywords— Data Mining, Classification, Infinitely Imbalanced Logistic Regression, Logistic Regression, Decision tree, KDD, Data Bases, likelihood, Machine Learning

I. INTRODUCTION

In many remote-sensing classification problems, the number of targets present is very little compared with the number of clutter objects encountered. For example, in land-mine detection applications, it is common to have nearly one hundred false alarms due to clutter for every real mine present. Similarly, in underwater-mine classification applications, the number of naturally occurring clutter objects such as rocks that are detected typically far outweighs the relatively rare event of detecting a mine.

Nevertheless, such class imbalance is often implicitly ignored when it comes time to choose a classification algorithm to use. As a result, the traditional classification approaches that do not account for severe class imbalance often lead to poor classification performance. A recently developed classification technique named infinitely imbalanced logistic regression explicitly acknowledges the problem of class disparity in its formulation. Although the method was developed with an eye toward applications involving rare events such as fraud detection or drug discovery, we apply the technique to the problem of mine classification in remote-sensing data. Specifically, we demonstrate the benefits of the proposed approach on three data sets of real, measured remote-sensing data. Various approaches have been developed to handle the issue of class imbalance in general. However, despite their relevance for many applications in the remote-sensing community, such methods have not been widely adopted. One notable exception to this sought to categorize agricultural crops in multi-spectral imagery by employing neural networks that were specially designed for class imbalance. A second example uses an ad hoc method for detecting oil spills in synthetic aperture sonar imagery. Our work is the first to address the matter of class imbalance for mine classification.

II. DATA MINING OVERVIEW

Data mining [Chen et al. 1996] is the process of extracting interesting (non-trivial, implicit, previously unknown and potentially useful) information or patterns from large information repositories such as: relational database, data warehouses, XML repository, etc. Also data mining is known as one of the core processes of Knowledge Discovery in Database (KDD).

Many people take data mining as a synonym for another popular term, Knowledge Discovery in Database (KDD). Alternatively other people treat Data Mining as the core process of KDD. The KDD processes are shown in Figure 2.1 [Han and Kamber 2000]. Usually there are three processes. One is called preprocessing, which is executed before data mining techniques are applied to the right data. The pre- processing includes integration, data cleaning, selection and transformation. The main process of KDD is the data mining process, in these process diverse algorithms are applied to produce concealed knowledge. After that comes another process called post processing, which evaluates the mining result according to users' requirements and domain knowledge? Regarding the evaluation results, the knowledge can be presented if the result is satisfactory, otherwise we have to run some or all of those processes again until we get the satisfactory result. The in fact processes work as follows.

First we need to clean and integrate the databases. Since the data source may come from different databases, which may have some inconsistencies and duplications, we must refresh the data source by removing those noises or make some compromises. Let us consider two different databases, different words are used to refer the same thing in their schema. When we try to integrate the two sources we can only choose one of them, if we know that they denote the same thing. And also real world data tend to be unfinished and noisy due to the manual input mistakes. The integrated data basis can be stored in a exclusive database, data warehouse or other repositories. As not all the data in the database are related to our mining task, the second process is to select task related data from the integrated resources and transform them into a format that is ready to be mined. Suppose we want to discover which items are often purchased together in a supermarket, while the database that records the purchase history may contains items bought, customer ID, transaction time, prices, number of each items , but for this specific task we only require items bought. After selection of relevant data, the database that we are going to apply our data mining techniques to will be much smaller, consequently the whole process will be more efficient.

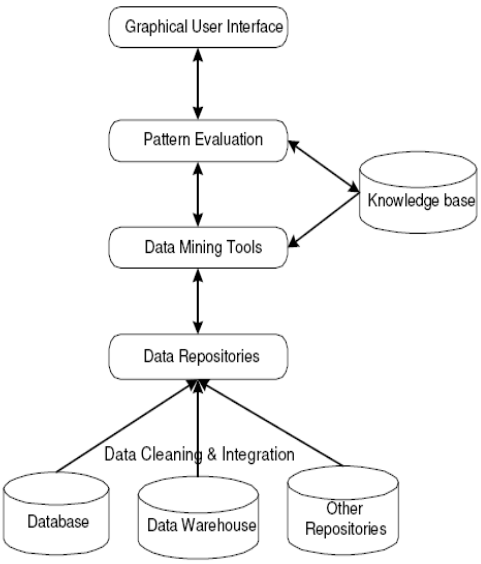

Fig. 2.1. Knowledge Discovery in Database processes

Various data mining techniques are applied to the data source, different knowledge comes out as the mining result. That knowledge is evaluated by certain rules, such as the domain knowledge or concepts. After the evaluation, as shown in Figure 3, if the result does not satisfy the requirements or contradicts with the domain knowledge, we have to redo some processes until getting the right results. Depending on the evaluation result we may have to redo the mining or the user may adjust his requirements. After we get the knowledge, the final step is to visualize the results. They can be displayed as raw data, tables, decision trees, rules, charts, data cubs or 3D graphics. This process is tried to make the data mining results easier to be used and more understandable.

III. TYPES OF MINING

Generally speaking, there are two classes of data mining descriptive and prescriptive. Descriptive mining is to summarize or characterize general properties of data in data repository, while prescriptive mining is to carry out inference on current data, to make predictions based on the past data. There is range of data mining techniques such as association rules, classifications and more importantly clustering. Based on those techniques web mining and sequential pattern mining are also well researched. We will review those different types of mining techniques with examples. Association Rule Association rule mining, one of the most important and well researched techniques of data mining, was first introduced in [Agrawal et al. 1993]. It aims to take out interesting correlations, frequent patterns, associations or casual structures among sets of items in the transaction databases or other data repositories. Example1.1. In an online book store there are always some tips after you purchase some books, for instance, once you bought the book Data Mining Concepts and Techniques, a list of related books such as: Database System 40%, Data Warehouse 25%, will be presented to you as recommendation for further purchasing.

In the above example, the association rules are: when the book Data Mining Concepts and Techniques is brought, 40% of the time the book Database System is brought together, and 25% of the time the book Data Warehouse is brought jointly. Those rules discovered from the transaction database of the book store can be used to reorganize the way of how to place those related books, which can further make those rules more strong. Those rules can also be used to help the store to make his market strategies such as: by upgrading the book Data Mining Concepts and Techniques, it can blows up the sales of the other two books mentioned in the example. Association rules are widely used in various areas such as telecommunication networks, market and risk management, inventory control etc. Various association mining techniques and algorithms will be briefly introduced and compared later in this Chapter. Classification [Han and Kamber 2000] is to build (automatically) a model that can catalogue a class of objects so as to predict the classification or missing attribute value of future objects (whose class may not be known). It is a two-step process. In the first process, based on the collection of training data set, a model is build to describe the characteristics of a set of data classes or concepts. Since data classes or concepts are predefined, this step is popularly also known as supervised learning (i.e., which class the training sample belongs to is provided). In the second step, the model is used to predict the classes of future objects or data. There are handful techniques for classification [Han and Kamber 2000]. Classification by decision tree was well researched and plenty of algorithms have been designed, Murthy did a comprehensive survey on decision tree induction [Murthy 1998]. Bayesian classification is another technique that can be found in Duda and Hart [Duda and Hart 1973]. Nearest neighbor methods are also discussed in many statistical texts on classification, such as Duda and Hart [Duda and Hart 1973] and James [James 1985]. Many other machine learning and neural network techniques are used to help constructing the classification models. A typical example of decision tree is shown in Figure 2. A decision tree for the class of buy a laptop, point out whether or not a customer is likely to purchase a laptop. Each internal node represents a

decision based on the value of corresponding attribute, also each leaf node represents a class (the value of buy laptop =Yes or No). After this model of buy laptop has been built, we can predict the likelihood of buying laptop based on a new customer's attributes such as age, degree and profession.

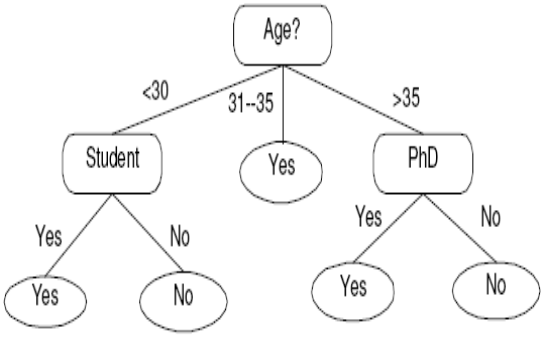

Fig. 2.2. An example of decision tree

That information can be used to target customers of certain products or services, especially widely used in insurance and banking. Clustering as we mentioned before, classification can be taken as supervised learning process, clustering is another mining technique alike to classification. However it is evident that clustering is an unsupervised learning process. Clustering is the process of grouping a set of physical or abstract objects into classes of similar objects [Han and Kamber 2000], so that objects within the same cluster must be similar to some extent, also they should be dissimilar to those objects in other clusters. In classification which record belongs which class is predefined, while in clustering there is no predefined classes. In clustering, objects are grouped together based on their similarities. Similarities among objects are defined by similarity functions, typically similarities are quantitatively specified as distance or other measures by corresponding domain experts.

Most clustering applications are used in market segmentation. By clustering their customers into different groups, business organizations can provide different personalized services to different group of markets. For example, based on the expense, deposit and draw patterns of the customers, a bank can clustering the market into different groups of people. For different groups of market, the bank can provide different kinds of loans for houses or cars with different budget plans. In this case the bank can provide a better service, and also make sure that all the loans can be reclaimed. A comprehensive survey of current clustering techniques and algorithms is available in [Berkhin 2002].

DATA CLASSIFICATION

Data mining involves the use of sophisticated data analysis tools to discover previously unknown, valid patterns and relationships in large data set. These tools can include statistical models, mathematical algorithm and machine learning methods. Therefore, data mining is process of more than collection and managing data, it also includes analysis and prediction. Classification technique is competent of processing a wider variety of data than regression and is growing in popularity.

There are several applications for Machine Learning (ML), the most significant of which is data mining. People are often prone to making mistakes during analyses or, perhaps, when trying to establish relationships between multiple features. This makes it difficult for them to find solutions to certain problems. Machine learning can often be successfully applied to these problems, improving the efficiency of systems and the designs of machines.Numerous ML applications involve tasks that can be set up as supervised. In the present paper, we have concentrated on the techniques necessary to do this. In particular, this work is concerned with classification problems in which the output of instances admits only discrete, unordered values.

DECISION TREE INDUCTION

Decision trees are trees that classify instances by sorting them based on feature values. Each node in a decision tree represents a feature in an instance to be classified, and each branch represents a value that the node can assume. Instances are classified starting at the root node and sorted based on their feature values. An example of a decision tree for the training set of Table I.

At1	At2	At3	At4	class
a1	a2	a3	a4	yes
a1	a2	a3	b4	yes
a1	b2	a3	a4	yes
a1	b2	b3	b4	no
a1	c2	a3	a4	yes
a1	c2	a3	b4	no
b1	b2	b3	b4	no
c1	b2	b3	b4	no

Figure 2.3 Training Data Set

Using the decision tree as an example, the instance At1 = a1, At2 = b2, At3 = a3, At4 = b4 would sort to the nodes: At1, At2, and finally At3, which would classify the instance as being positive (represented by the values "Yes"). The problem of constructing optimal binary decision trees is an NP complete crisis and thus theoreticians have searched for efficient heuristics for constructing near-optimal decision trees.The feature that best divides the training data would be the root node of the tree. There are numerous methods for finding the feature that best divides the training data such as information gain (Hunt et al., 1966) and gini index (Breiman et al., 1984). While myopic measures estimate each attribute independently, ReliefF algorithm (Kononenko, 1994) estimates them in the context of other attributes. However, a majority of studies have concluded that there is no single best method (Murthy,1998). Comparison of individual methods may still be important when deciding which metric should be used in a particular dataset. The identical procedure is then iterated on each partition of the divided data, creating sub-trees until the training data is divided into subsets of the same class.

The basic algorithm for decision tree induction is a greedy algorithm that constructs decision trees in a top-down recursive divide-and-conquer manner. The algorithm, summarized as follows.

create a node N;
if samples are all of the same class, C then
return N as a leaf node labeled with the class C;
if attribute-list is empty then
return N as a leaf node labeled with the most
common class in samples;
select test-attribute, the attribute among attribute-
list with the highest information increase;
label node N with test-attribute;
for each known value ai of test-attribute
grow a branch from node N for the condition test-
attribute= ai;
let si be the set of samples for which test-attribute=
ai;
if si is empty then
attach a leaf labeled with the most common class
in samples;
else attach the node returned by
Generate_decision_tree(si,attribute-list_test-
attribute)

Fig.2.4 Algorithm for a decision tree

A decision tree, or any learned hypothesis h, is said to over fit training data if another hypothesis h□ exists that has a larger error than h when tested on the training data, but a smaller error than h when tested on the entire dataset. There are two common approaches that decision tree induction algorithms can use to shun over fitting training data: i) Stop the training algorithm before it reaches a point at which it perfectly fits the training data, ii) Prune the induced decision tree. If the two trees are employing the same kind of tests and have the same prediction correctness, the one with less leaves is usually preferred. Breslow & Aha (1997) survey methods of tree simplification to improve their comprehensibility.

Decision trees are usually unvaried since they use based on a single feature at each internal node. Most decision tree algorithms cannot perform well with problems that require diagonal partitioning. The division f the instance space is orthogonal to the axis of one variable and parallel to all other axes. Consequently, the resulting regions after partitioning are all hyper rectangles. However, there are a few methods that construct multivariate trees. One example is Zheng's(1998), who improved the classification accuracy of the decision trees by constructing new binary features with logical operators such as conjunction, negation, and disjunction. In addition, Zheng (2000) created at-least M-of -N features. For a given instance, the value of an at least M-of -N representation is true if at least M of its conditions is true of the instance, otherwise it is false. Gama and Brazdil (1999) combined a decision tree with linear discriminate for constructing multivariate decision trees. In this model, additional features are computed as linear combinations of the previous ones.

Decision trees can be significantly more complex representation for some concepts due to the replication problem. A solution is using an algorithm to implement complex features at nodes in order to avoid replication. Markovitch and Rosenstein (2002) presented the FICUS construction algorithm, which receives the standard input of supervised learning as well as a feature representation specification, and uses them to produce a set of generated features. While FICUS is similar in some aspects to other feature construction algorithms, its main strength is its generality and flexibility. FICUS was designed to perform feature generation given any feature representation specification complying with its general purpose grammar. The most well-know algorithm in the literature for building decision trees is the C4.5 (Quinlan, 1993).

C4.5is an extension of Quinlan's earlier ID3 algorithm (Quinlan, 1979). One of the latest studies that compare decision trees and other learning algorithms has been done by (Tjen-Sien Lim et al. 2000). The study shows that C4.5 has a very good combination of error rate and speed. In 2001, Ruggieri presented an analytic evaluation of the runtime behavior of the C4.5 algorithm, which highlighted some efficiency improvements. Based on this analytic evaluation, he implemented a more efficient version of the algorithm, called EC4.5. He argued that his implementation computed the same decision trees asC4.5 with a performance gain of up to five times.

To sum up, one of the most useful characteristics of decision trees is their comprehensibility. People can easily understand why a decision tree classifies an instance as belonging to a specific class. Since a decision tree constitutes a hierarchy of tests, an unknown feature value during classification is usually dealt with by passing the example down all branches of the node where the unknown feature value was detected, and each branch outputs a class distribution. The output is a blend of the different class distributions that sum to 1. The assumption made in the decision trees is that instances belonging to different classes have different values in at least one of their features. Decision trees tend to perform superior when dealing with discrete/categorical features.

Logistic Regression (LR) :

Logistic regression (LR) is a commonly used approach for performing binary classification. It learns a set of parameters that maximizes the likelihood of the class labels for a given set of training data. Column vector denotes several features representing the data points, as well as denoting its corresponding class label (e.g., clutter or mine). For a labeled training data point, y is known; for an unlabeled testing data point, y is unknown. For a set of N independent, labeled data points, maximum log-likelihood of the class labels is achieved through; a standard optimization approach can be employed, since the gradient can be calculated readily. Once the logistic-regression parameters have been learned, the probability that an unlabeled testing data point x belongs to each class can be obtained.

Infinitely Imbalanced Logistic Regression (IILR):

When the number of data points belonging to one class far exceeds the number belonging to the other class, the standard logistic regression approach can lead to poor classification performance. This scenario of class imbalance motivates the use of a modified form of logistic regression named infinitely imbalanced logistic regression. Let N1 and N0 be the number of labeled data points belonging to classes y = 1 and y = 0, respectively. Let the ith labeled data point belonging to class y = 1 be written as x1; similarly, write the ith labeled data point belonging to class y = 0 as x0. With this notation, the log-likelihood can be re-written as without loss of generality, assume that N0 is very large so that replacing the first sum by an integral is justifiable. Then write the average of the class y = 1 data points. The limiting value of w that maximizes is found to satisfy

IV. CONCLUSIONS

This work proposes, a recently-developed classification approach for data sets characterized by class imbalance has been described. The implementation details needed to employ the approach have also been provided. The utility of the infinitely imbalanced logistic regression algorithm for mine classification was demonstrated on E collision data set. Many classification algorithms in use today are relatively similar to logistic regression. Therefore, after introducing this method to the community, I expect that other researchers will find the approach helpful for other applications characterized by severe class imbalance.

REFERENCES

[1] A. Owen, “Infinitely imbalanced logistic regression,” Journal of Machine Learning Research, vol. 8, pp. 761–773, 2007.
[2] L. Bruzzone and S. Serpico, “Classification of imbalanced remotesensing data by neural networks,” Pattern Recognition Letters, vol. 18, pp. 1323–1328, 1997.
[3] M. Kubat, R. Holte, and S. Matwin, “Machine learning for the detection of oil spills in satellite radar images,” Machine Learning, vol. 30, pp. 195–215, 1998.
[4] A. Dempster, N. Laird, and D. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” J. Royal Statistical Society B, vol. 39, pp. 1–38, 1977.
5] N. Nasios and A. Bors, “Variational expectation-maximization training for Gaussian networks,” in Proceedings of the IEEE Workshop on Neural Networks for Signal Processing, 2003, pp. 339–348.
[6] M. Beal and Z. Ghahramani, “The variational Bayesian EM algorithm for incomplete data: Application to scoring graphical model structures,” Bayesian Statistics, vol. 7, pp. 453–464, 2003.
[7] V. Myers and Ø. Midtgaard, “Fusion of contacts in synthetic aperture sonar imagery using performance estimates,” in IOA International Conference on Detection and Classification of Underwater Targets, 2007, pp. 77–88.

